



ELSEVIER

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Computer Networks 44 (2004) 757–772

COMPUTER
NETWORKS

www.elsevier.com/locate/comnet

Admission control by implicit signaling in support of voice over IP over ADSL

Abhishek Ram^a, Luiz A. DaSilva^{b,*}, Srinidhi Varadarajan^a

^a Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA

^b Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, 206 N Washington Street, Suite 400, Alexandria, VA 22314, USA

Received 18 September 2002; received in revised form 24 June 2003; accepted 4 November 2003

Responsible Editor: K. Park

Abstract

Voice over DSL (VoDSL) is a technology that enables the transport of data and multiple voice calls over a single copper-pair. Voice over ATM (VoATM) and Voice over IP (VoIP) are the two main alternatives for carrying voice over DSL. ATM is currently the preferred technology, since it offers the advantage of ATM's built-in Quality of Service (QoS) mechanisms. IP QoS mechanisms have been maturing only in recent years. However, if VoIP can achieve comparable performance to that of VoATM in the access networks, it would facilitate end-to-end IP telephony and could result in major cost savings. In this paper, we propose a VoIP-based VoDSL architecture that provides QoS guarantees comparable to those offered by ATM in the DSL access network. Our QoS architecture supports Premium and Regular service categories for voice traffic and the Best-Effort service category for data traffic. The Weighted Fair Queuing algorithm is used to schedule voice and data packets for transmission over the bottleneck link. Fragmentation of large data packets reduces the waiting time for voice packets in the link. We also propose a new admission control mechanism called Admission Control by Implicit Signaling. This mechanism takes advantage of application layer signaling by mapping it to the IP header. We evaluate the performance of our QoS architecture by means of a simulation study. Our results show that our VoIP architecture can provide QoS comparable to that provided by the VoATM architecture.

© 2003 Elsevier B.V. All rights reserved.

1. Motivation

Quality of service (QoS) remains an important issue for access networks. While in core networks

over-provisioning may be used to ensure adequate performance to real and non-real time applications, this is typically not feasible in access networks. In this paper, we propose a simple method to support service differentiation between voice and data flows in Asymmetric Digital Subscriber Line (ADSL).

Digital Subscriber Line (DSL) refers to a collection of technologies used for the transmission of high-speed data over copper twisted-pair lines. It is

* Corresponding author.

E-mail addresses: abhisram@windows.microsoft.com (A. Ram), ldasilva@vt.edu (L.A. DaSilva), srinidhi@cs.vt.edu (S. Varadarajan).

used to connect network service providers (NSP) and customers, typically residences or small-to-medium sized businesses. The Customer Premises Equipment (CPE) connects (via DSL) to a DSL Access Multiplexer (DSLAM) located in the Central Office (CO) of the NSP. The DSLAM aggregates traffic from different customers and sends it over high-speed links towards the core of the network. Fig. 1 illustrates the network topology of the DSL access network. Asymmetric Digital Subscriber Line (ADSL) is currently the most widely deployed DSL technology. ADSL provides higher bandwidth from the NSP to the customer (downstream) than from the customer to the NSP (upstream). It supports high data-rate services such as high-speed Internet access and streaming audio/video.

As an alternative to the circuit-switched services provided by the Public Switched Telephone Network (PSTN), voice over DSL (VoDSL) technology uses the existing DSL access network to provide voice services in addition to data services. Voice is packetized at the customer premises and the DSL access network is used to deliver the voice packets to a voice gateway. The voice gateway then converts packetized voice into circuit-switched voice traffic and sends it to the PSTN. This enables a single copper pair to provide data services and simultaneously support several voice calls, reducing the costs of provisioning multiple copper pairs.

Voice over Asynchronous Transfer Mode (VoATM) has been proposed as the VoDSL technology of choice [9–11]. In the VoATM architecture, voice and data are carried over different virtual circuits (VCs) from the CPE to the DSLAM, which basically acts as an ATM switch.

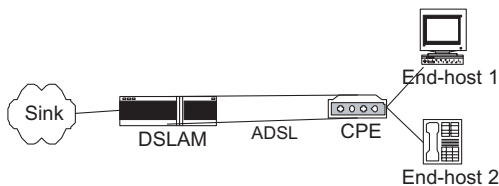


Fig. 1. Topology of the DSL Access Network. Our study focuses on the access network, and the core network (cloud in the figure) is treated as a sink.

Voice is carried directly over ATM Adaptation Layer 2 (AAL2) using Constant Bit Rate (CBR) service. Data are carried over IP over ATM Adaptation Layer 5 (AAL5) using Unspecified Bit Rate (UBR) service. The voice and data protocol stacks are shown in Fig. 2. At the DSLAM, AAL2 voice traffic is switched to a voice gateway and AAL5 data traffic is switched towards the core IP network. AAL2 imposes very low protocol overhead on voice packets and it also supports multiplexing of different voice flows over a single VC. VoATM also reaps the benefits of ATM’s built-in QoS support. The voice flows that get admitted are guaranteed a portion of the bandwidth and hence tend to experience low delays across the network. For the reasons discussed above, VoATM has gained favor with some proponents of VoDSL.

An alternative technology for carrying voice traffic over DSL is Voice over IP (VoIP) [9,10,31]. In this architecture, both voice and data traffic is carried over IP from the CPE to the DSLAM. Fig. 3 shows the protocol stack. The DSLAM acts as an IP router, routing data packets towards the core. If the PSTN is used for voice calls, then the DSLAM routes voice packets towards a voice gateway that converts packet voice into circuit-switched voice. If

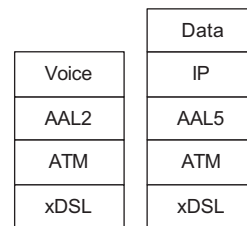


Fig. 2. Protocol stacks in VoATM.

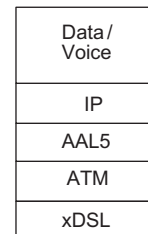


Fig. 3. Protocol stack for VoIP.

IP telephony is employed, then voice packets are also sent over the packet network towards the core. The Layer 2 protocol in the VoIP protocol stack may be ATM, Frame Relay or Serial Line IP (SLIP). In VoIP deployments, the Layer 2 protocol treats voice and data packets alike and typically does not perform any QoS-related functions. To reduce packetization delay and transmission delays to a minimum, it is best to keep the payload size of voice packets small. However, the VoIP protocol stack has several layers, each of which adds a header to the voice payload. These headers impose significant protocol overhead on the small voice packets. Solutions such as header compression [7,8,17], and IP trunking [9,15] (also known as RTP multiplexing) help alleviate the overhead problem.

Competition from data traffic poses a challenge for VoIP. During congestion, voice packets could get queued behind several large data packets at the routers, experiencing significant queuing delay. Placing voice packets in a separate queue and employing some scheduling algorithm is a possible solution to this problem. However, even with scheduling, a voice packet that arrives while a data packet is being transmitted has to wait for the transmission to be completed. The waiting time for voice packets could be anywhere between zero and the transmission time for the entire packet. Thus, voice traffic may experience significant variation in end-to-end delay. Fragmentation of large data packets can alleviate this problem.

There is considerable interest in using an all-IP core network capable of carrying both voice and data. If the performance of VoIP in the DSL access network can be improved using IP QoS mechanisms, then the deployment of VoIP for VoDSL would yield several benefits. First, the voice packet format used in the DSL access network would be compatible with that used in the core. This eliminates the need for a voice gateway and also paves the way for end-to-end IP telephony. Moreover, VoIP can run over any kind of DSL network—ATM based, frame-based or PPP-based. Thus, a solution that achieves VoIP performance comparable to that provided by VoATM would be of significant benefit in the DSL access network.

The primary contribution of this paper is to propose a mechanism for admitting and differentiating between voice and data flows in a DSL access network *without the need for explicit signaling*. We also evaluate the performance of our proposed mechanism for voice and data convergence over ADSL. Our results indicate that, by making use of implicit signaling as proposed here, it is possible to support voice over IP (VoIP) in an ADSL environment with quality comparable to that obtained with voice over ATM (VoATM). Also of note is that the form of signaling we propose imposes minimal additional processing requirements and no additional control traffic.

2. An IP QoS architecture for VoDSL

In this section, we describe our architecture, designed to support three service classes:

(i) *Premium Voice*—This service class is meant for voice calls that require excellent QoS. A percentage of the upstream bandwidth is reserved for premium voice calls. Admission control is strictly enforced for calls belonging to this category. Premium calls are not allowed to borrow unused bandwidth that is reserved for other classes of service, in order to avoid the stolen bandwidth problem (as discussed in Breslau et al. [6]). Premium traffic load must never be allowed to exceed the limit that can be supported by the bandwidth reserved for it.

(ii) *Regular Voice*—This service class is meant for voice calls that do not get admitted to the premium category. If we were to have only the premium category for voice calls, then voice calls would automatically get rejected if all the bandwidth reserved for the premium class were in use. Even if the bandwidth reserved for other classes of traffic were unused, we would be unable to take advantage of it, since premium traffic cannot borrow bandwidth from other classes. Such a strict fragmentation of resources leads to inefficient utilization. To address this issue, we introduce the concept of regular voice calls. Voice calls that cannot be admitted as premium are allowed to go through as regular calls. However, the regular calls are not given any QoS guarantees and share the

residual bandwidth with best-effort traffic (described next). There is no admission control for the regular voice traffic. Under conditions of light load from best-effort traffic, the regular calls will receive acceptable performance. In the case of ADSL, the typical customer runs applications such as web browsing, email, remote login and streaming multimedia, which offer low load on the upstream link. Besides, the downstream bandwidth is usually quite high, so there is a reasonable chance of a regular call experiencing acceptable quality in a lightly loaded network.

For simplicity, the classification of voice calls into regular and premium is done in a first-come-first-served basis. Alternatively, to increase fairness, the CPE could keep a history of recently admitted calls and use this information in the classification of new voice calls.

(iii) *Best Effort*—This service category is for non real-time traffic. Applications such as web browsing, email, FTP, remote login and database access fall under this category.

To prevent starvation of data flows, we do not allow premium voice flows to reserve the entire link bandwidth. Best effort and regular voice flows compete for the unreserved bandwidth.

At the interface of the CPE that forwards packets to the DSLAM, we have a separate queue for each of the service classes. The weighted fair queuing (WFQ) [12] algorithm is used to service these queues. The arrival of packets to the premium voice queue is controlled by an admission control mechanism described later.

WFQ scheduling is able to guarantee a portion of the upstream link bandwidth to each class. However, the system does not allow the preemption of packets already in transmission. This can pose a problem on low-speed links. A voice packet that arrives just after a large data packet has begun transmission will have to wait until transmission of the data packet is completed. This delay can be unacceptable for real-time applications. Depending on when they arrive, the waiting time for voice packets can vary between zero and the time taken to transmit the largest possible packet. Hence, the delay variation (or jitter) of the voice packets is also increased due to large data packets.

The simplest solution to this problem is to fragment large data packets. This can be done at the IP layer by reducing the Maximum Transmission Unit (MTU) for the interface over which the packets need to be sent. The solution is effective enough in reducing the delay and jitter of voice packets, but there are some undesirable side effects. The IP header is replicated in each of the fragments, increasing protocol overhead. Furthermore, fragmentation at the IP layer has an end-to-end effect, because after the packet is fragmented, it will be reassembled only at the destination. Another disadvantage is that applications that use PathMTU Discovery [21] will not send packets that are larger than the MTU. Hence, if the MTU is made very small, then application packets will be able to carry hardly any payload.

A better solution for this problem is obtained by using the Point-to-Point Protocol (PPP) [27]. PPP can be used over the ADSL link connecting the CPE and the DSLAM. Multilink-PPP (MP) [28] is a PPP variant that fragments PPP packets before sending them over the link. Caputo [7] and Armitage [1] describe a solution based on MP fragmentation to reduce the delay and jitter of voice packets. The idea is to have the upstream link shared by ‘IP-over-MP’ and ‘IP-over-PPP’ running simultaneously. The low priority data packets (belonging to the best-effort class) are forwarded using ‘IP-over-MP’ and the high priority voice packets are forwarded over ‘IP-over-PPP’. This is shown in Fig. 4. Since MP forces fragmentation of the packets, the transmission of voice packets could be interleaved between the transmissions of the MP fragments for a single data packet. This eliminates the delay and jitter problem described earlier and ensures that voice packets are serviced in a timely manner. Since fragmentation is done at the data link layer, the IP header is not replicated in each packet. Moreover, the fragmentation does not have an end-to-end effect, since the fragments are reassembled at the other end of the access link.

2.1. Admission control by implicit signaling (ACIS)

We propose an admission control mechanism based on an implicit signaling protocol. We call

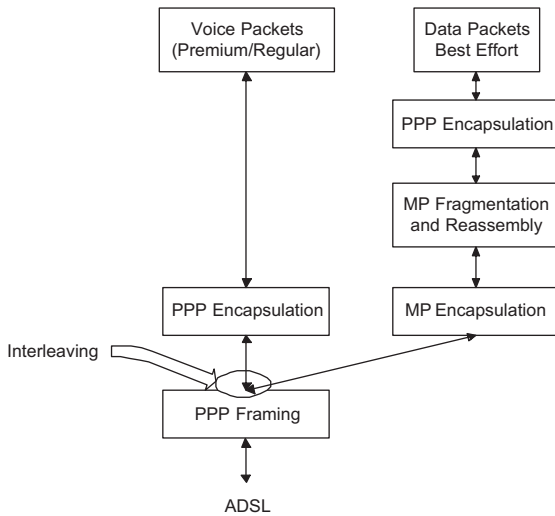


Fig. 4. MP fragmentation (adapted from [1]).

our mechanism Admission Control by Implicit Signaling (ACIS). ACIS is implemented in the CPE to limit the number of voice calls that are admitted to the premium category. This mechanism takes advantage of signaling by application layer protocols such as the Session Initiation Protocol (SIP) [26] or H.323. In this document, we use SIP to explain our idea. The ACIS mechanism does not reject any voice calls. Instead, it admits only a limited number of voice calls as premium voice calls and admits the rest as regular voice calls. In this manner, we limit the arrival rate of premium traffic and guarantee low latency for premium calls. On the other hand, there is no limit to the number of voice calls that can be admitted as regular calls. Hence, these calls may receive satisfactory service at light loads, but the quality degrades as the load increases.

2.1.1. Encoding application layer signaling information

The key idea in ACIS is to map the application layer signaling protocol onto the IP header, so that the CPE can infer the setup and teardown of voice connections by looking at the IP header. For the working of ACIS, the IP header must incorporate the additional fields shown in Fig. 5. We refer to these fields collectively as the ACIS fields.



Fig. 5. ACIS fields in the IP header.

The Packet Type (PT) field indicates whether the packet contains an application layer signaling message, or just voice payload. If the packet contains an application layer signaling message, then the type of message is also encoded in the PT field. The Encoder (ENC) field represents the voice encoding scheme used by the application. Finally, the Number of Frames (NFR) field contains the number of voice frames contained in the voice packet. The lengths of the above fields are implementation dependent. The TOS octet in the IP header can contain these fields. However, depending on the implementation, it is possible that the eight bits of the TOS field would be insufficient to encode all the information.

An alternative approach would be to encode the ACIS fields onto the Offset field of the IP header. A similar approach is adopted in the DPS technique proposed in Stoica and Zhang [29]. Since voice packets are typically small, it is highly unlikely that they would ever be fragmented. Hence, the Offset field can be used to store some of the above information, if necessary. If this is done, a flag must be set in the IP header to prevent the CPE from interpreting the field as a true fragment offset value.

Note that the ACIS fields have significance within the customer premises only. Before forwarding the packet towards the DSLAM, the CPE will mark the TOS fields in accordance with whatever QoS scheme is in use outside the customer premises. If the Offset field of the IP header is being used to hold ACIS information, then the CPE will zero the Offset field before forwarding the packet. In that manner, core routers are oblivious to the use of the TOS and/or Offset fields for traffic differentiation in the access network.

The bandwidth required by voice calls can be inferred from the ENC and NFR fields, if a standard encoder is used. The encoding rates and frame sizes of the standard encoders are well known. The NFR field contains the number of frames included per voice packet. Using the frame

size and the number of frames per packet, we can compute the payload size for each packet. The total header size can also be computed if the protocol stack is known. Using the header and payload sizes, we can calculate the protocol overhead imposed on the voice packets. Finally, we scale the encoding rate according to the protocol overhead to obtain the actual bandwidth requirement of the connection. For example, the G.729 encoder produces a voice stream at the rate 8 kbps. The frame size of G.729 is 10 ms. When expressed in bits, the frame size is $(8 \times 1000) \times (10/1000) = 80$ bits. If we have two voice frames per packet, then the voice payload in a packet is $2 \times 80 = 160$ bits. Assume a protocol stack in which the protocols together add a header of size 40 bytes (320 bits). Then, the total packet size becomes $160 + 320 = 480$ bits. Thus, for every 160 bits of voice payload, there are 320 bits of header. The total bandwidth requirement, in this case, is $8 + 16 = 24$ kbps.

Instead of having the ENC and NFR fields, an alternative approach would be to have the application itself compute the bandwidth requirement and encode it in the header. However, this would require the application to perform the computations described above. We have chosen to use the ENC and NFR fields because it simplifies the application by shifting the burden of the computing the bandwidth from the application to the CPE. With this approach, we also spare the application from having to understand the details of the underlying protocol stack. Of course, it is possible to combine the two approaches by using a special ENC value to indicate to the CPE that the application has computed the bandwidth and encoded it in another field (say, BW). When the CPE sees this special value, it would directly read the bandwidth from the BW field, instead of attempting to compute the bandwidth. To simplify our discussion, the rest of the paper assumes that the CPE computes the bandwidth using the ENC and NFR fields.

2.1.2. Hash table description

The CPE maintains connection information for each premium call that it admits. It uses a hash table for this purpose. As we shall see later, the CPE needs to search the list of premium calls

frequently; hence a fast method for looking up this information is desirable. The hash table was chosen as the data structure in order to enable the premium call information to be located in $O(1)$ time.

However, it is important to ensure that the computations associated with the hashing function are not so expensive that they nullify the benefit offered by the $O(1)$ lookup. If we assume that each end host participates in only one call at any given time, then we can use the IP address of the end-host as the key for the hashing function. Further, if we assume that the IP addresses of the end-hosts in the customer premises are contiguous, then we can employ a simple hashing function. If the smallest IP address in the end host is called Base, then for any end host, the value (IP Address—Base) can be used as an index into the hash table. The computation overhead incurred by such a hashing function is minimal. Moreover, with this scheme, the size of the hash table will also be bounded by the maximum number of end-hosts in the customer premises, which is likely to be small.

Though we have chosen a hash table as the data structure, it should be noted that ACIS is not tightly coupled with the data structure used to store information about premium calls. ACIS will require very minor modifications in order to work with other data structures.

The hash table has the following fields:

- (i) Premium Flag—This is a flag that indicates whether the call is admitted as premium. This flag determines the validity of a hash table entry. If the flag is set, the hash table entry is valid.
- (ii) Bandwidth—This field stores the bandwidth requirements of the voice call.
- (iii) Attempts—This field stores the number of connection attempts made by this call.
- (iv) Last Used—This field stores the time when a packet belonging to this call last arrived at the CPE.

2.1.3. ACIS operation

Fig. 6 illustrates the operation of ACIS. When a SIP-based application wishes to establish a VoIP call, it sends an “Invite” message to the intended

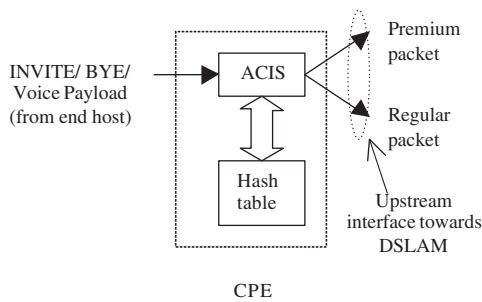


Fig. 6. ACIS operation.

receiver. The ACIS fields in the IP header of the “Invite” message are used to alert the CPE to the establishment of a new connection. They also convey the bandwidth requirements to the CPE, as described above. The CPE keeps track of the total bandwidth already reserved for premium calls and the bandwidth that is currently available for incoming calls. The CPE tries to accommodate every incoming call as a premium call and classifies calls into the regular category only when sufficient bandwidth is not available. This classification of calls into premium and regular by the CPE is transparent to the end-hosts. The end-hosts are not explicitly informed whether their call has been accepted as premium or regular.

Upon receiving the “Invite” message, the CPE first checks whether this is the first connection attempt being made by the end-host. This is done by checking whether the hash table has a valid entry for this call. If this is the first attempt, then the CPE computes the bandwidth required for this call and compares it against the bandwidth availability for premium calls. If sufficient bandwidth is available, it admits the call as a premium voice call. The following updates occur in the hash table entry for that call:

- The ‘Premium’ flag is set.
- The bandwidth requirement is stored in the ‘Bandwidth’ field.
- Since this is the first connection attempt, the value in the ‘Attempts’ field is set to 1.
- The ‘Last Used’ field is set to the current time.

If sufficient bandwidth is not available in the premium category or if the CPE is not able to

compute the bandwidth requirement for the call because it does not recognize the encoding scheme specified in the ENC field, then the call is admitted as a regular voice call.

The packet is placed in the appropriate queue according to its class of service. Optionally, the CPE may perform further marking of the IP datagram according to some QoS marking that is recognized beyond the access network. The ACIS fields have significance only within the customer premises.

It is possible that the CPE receives duplicate “Invite” messages belonging to the same connection. This may be caused by retransmission due to packet loss or it may be a retry if the connection was rejected in a previous attempt. Thus, each time the CPE receives an “Invite” message, it checks the hash table to determine whether the call has already been admitted as a premium call. If so, then the CPE does not attempt to allocate more bandwidth for that call. Instead, it just increments the number of connection attempts, stored in the ‘Attempts’ field of the hash table. If the number of connection attempts exceeds a fixed maximum, then the CPE revokes the premium status of this call. It resets the ‘Premium’ flag in the hash table and treats the “Invite” packet as a regular category packet. This prevents a call from occupying the premium list for too long without being able to establish a connection.

The “Invite” message is forwarded as usual through the network to the receiver, which would then confirm the acceptance of the request by the transmission of an “ACK” message, establishing the session. After the session has been established, the end-host starts sending voice packets. The ACIS fields in the voice datagram are marked to indicate the packet type and bandwidth requirement. When the CPE receives a voice packet, it first verifies whether the packet belongs to a premium call. This is accomplished by checking whether the hash table has a valid entry for this call. If so, then the ‘Last Used’ field of the hash table is updated to the current time, and the packet is forwarded towards the DSLAM. However, if the voice packet does not belong to the premium call, the CPE checks whether the call can be upgraded from regular to premium category at this

stage. This might be possible, if some premium calls have terminated, thereby freeing some premium bandwidth, after this particular call was established. Note that if several regular calls are in progress and there is enough bandwidth to upgrade only one of those, then the call that gets upgraded is the one whose voice packet reaches the CPE first after the additional bandwidth becomes available. If sufficient bandwidth is not available to make an upgrade, then the packet is simply forwarded as belonging to a regular call.

When the end-host wishes to end the voice call, SIP sends a “Bye” message to indicate the end of session to the other party. As usual, the ACIS fields are marked to indicate the packet type and the bandwidth requirement. When the CPE intercepts this message, it checks the hash table to determine whether the packet belongs to a premium call. If so, the ‘Premium’ flag in the hash table is reset to indicate that the call has terminated. The bandwidth availability for premium calls is increased to reflect the closing of this session. The TOS value of the packet is marked to indicate that it belongs to a premium call and the packet is then forwarded. If the “Bye” message belongs to a regular call, then the CPE simply marks the packet as a regular packet and forwards it. The eventual arrival of a duplicate “Bye” message does not affect the correct operation of ACIS.

The mechanism described above is further refined to address robustness issues. It is possible for a premium voice application running on the end-host to crash in the middle of a call. In this case, the application would never send a “Bye” message to indicate the end of the call. The CPE needs some mechanism to de-allocate the bandwidth that it had reserved for this application. To deal with this, we use the concept of soft state. For each premium call, the CPE keeps track of the time when a packet from the call last arrived. This is stored in the ‘Last Used’ field in the hash table entry. ACIS sets a maximum silence period for premium calls. If a premium call has not sent a packet for a period longer than the maximum silence period, then the premium status of the call can be revoked. This could be done by the CPE on a periodic basis. Alternatively, it could be done on a need basis, i.e. if a new connection request ar-

rives and there is no bandwidth available for it. A possible implementation would involve a combination of the two choices. That is, the CPE could revoke the premium status of idle connections at longer intervals, while new connection requests could trigger an immediate check for idle connections whose premium status can be revoked. In either case, the CPE scans through the list of premium calls to see if there exist one or more calls that have not sent a packet for a period longer than the maximum silence period. Such calls are removed from the premium category in order to free some bandwidth to support incoming calls.

2.2. Related work

Having described our QoS architecture, we compare it with some similar research efforts seen in the literature. Breslau et al. [6], Bianchi et al. [2], Borgonovo et al. [4], Elek et al. [14] and Mase et al. [20] discuss endpoint admission control, a technique in which the end-hosts use probe packets to measure the QoS provided by the network. Based on these measurements, the end-hosts themselves make the admission control decisions. ACIS differs from endpoint admission control in that the admission control decision is made by the CPE and not the end-hosts. Moreover, ACIS does not operate on an end-to-end basis, it operates over the DSL access network only.

Measurement based admission control (MBAC) is a technique in which routers measure the traffic load in the network and make admission control decisions based on these. Rhee et al. [25], Bianchi et al. [3] discuss MBAC. A bandwidth broker (BB), which is an entity that manages resources for an entire network, can also be used to perform admission control. This idea is discussed in [13,16,18,19,30,32]. ACIS can operate in conjunction with MBAC or with BB. In this paper, it is assumed that bandwidth reserved for premium calls is fixed. However, the CPE could employ some MBAC scheme to perform some measurements on the traffic load and dynamically reserve bandwidth for premium calls based on these measurements. Alternatively, if a BB were to be used in the service provider’s network, the CPE could periodically consult the BB to determine

bandwidth availability. It could dynamically reserve bandwidth for premium calls based on this information from the BB. Even though the bandwidth reserved for premium calls is not a constant in either of these models, ACIS can still be used to make admission control decisions at the CPE.

Admission control often involves the use of an explicit signaling protocol such as Resource Reservation Protocol (RSVP) [5], Sender-initiated Resource Reservation Protocol (SRRP) [33] and DiffServ PHB Reservation Protocol (DPRP) [23]. In contrast to these, ACIS uses an implicit signaling protocol to perform admission control. Mortier et al. [22] proposes an admission control scheme which, like ACIS, uses the concept of a higher layer connection to perform admission control. The paper describes how routers can snoop TCP packets to identify SYN and SYN/ACK, which indicate connection establishment. The connection is then rejected or accepted by dropping the packet or allowing it to pass. ACIS differs from this scheme in that it does not involve looking into the higher layer information stored in the IP payload. Instead, it maps higher layer information on to the IP header. In addition, ACIS classifies connections into different service classes rather than merely accepting or rejecting them.

3. Simulation model

The system of interest in our study is the DSL access network, ranging from the end-hosts at the customer premises to a traffic sink located just beyond the DSLAM, as illustrated in Fig. 1. The key components of this system are the end-hosts, the CPE, the ADSL link, the DSLAM and the traffic sink. The metrics used to evaluate the performance of voice traffic are the end-to-end delay of voice packets (ETE Delay) *across the access network* and the bandwidth required for the voice calls.

For the VoATM scenarios, the end hosts consist of Ethernet-based clients generating data traffic and AAL2-based clients generating voice traffic. The CPE has an Ethernet interface for data traffic and AAL2 interfaces for voice traffic. The

interface connecting the CPE to the DSLAM is ATM-based. The DSLAM is an ATM switch.

For the VoIP scenarios, the end-hosts are all Ethernet based clients that can generate both voice and data traffic. The CPE is an IP router, which has an Ethernet interface for the all the traffic originating from the end-hosts. The interface connecting the CPE to the DSLAM could be ATM, SLIP, PPP or Frame Relay. The DSLAM is another IP router that routes the voice and data packet towards the traffic sinks.

We used OPNET ModelerTM for our simulations. OPNET ModelerTM supports several existing IP and ATM QoS mechanisms. However, in modeling the ACIS mechanism, we added functionality to the IP layer at the CPE and to the interface between the application layer and the transport layer at the end-hosts.

4. Performance evaluation

In this section, we describe the experiments performed to evaluate the performance of our system. *In all experiments discussed here, the ADSL link has upstream bandwidth of 128 kbps and downstream bandwidth of 1.5 Mbps.* The upstream ADSL link is the bottleneck in our system; all other links have sufficient capacity to serve the arriving packets with minimum queuing and transmission delays.

4.1. VoATM model

We begin by establishing baseline results for the latency experienced by voice packets in a VoATM architecture. We first perform an experiment with both voice and data receiving UBR service. We then observe the improvement in performance when voice receives CBR service and data receives UBR service.

In our experimental setup, we have four end-hosts in the customer premises engaged in voice calls. They use the G.729 encoder and include 2 voice frames in each packet. Each call requires 21.2 kbps, including protocol headers. One end-host on the customer premises generates data traffic by uploading a 2 MB file via FTP.

In our first experiment, both voice and data receive UBR service. Bandwidth guarantees are not given to any of the flows. The FTP application generates significant traffic that competes with voice traffic for the upstream bandwidth. Both voice and data traffic share the UBR queue. Since the voice calls are not guaranteed any portion of the bandwidth, the average ETE delay of the voice packets is high. Next, we repeat the experiment with voice traffic receiving CBR service and data traffic receiving UBR service. Using ATM's call setup mechanism, each voice call requests CBR service and specifies a bandwidth requirement of 21.2 kbps. Voice packets are sent to the CBR queue and data packets are sent to the UBR queue. The CBR queue has 50% of the upstream link bandwidth (50% of 128 kbps = 64 kbps) reserved for it. In this experiment, the 64 kbps reserved for voice in the upstream can support a maximum of three voice calls ($3 \times 21.2 \text{ kbps} = 63.6 \text{ kbps}$). Hence, the CPE rejects one of the four voice calls during call setup. However, the three accepted calls experience low ETE delays. Since the ATM cells are all small and equal in size, a cell that arrives in the CBR queue while a UBR cell is being transmitted will not have to wait long. Hence, the delay variation is low. The results are shown in Table 1. Fig. 7 shows the improvement in performance obtained by using CBR service for voice.

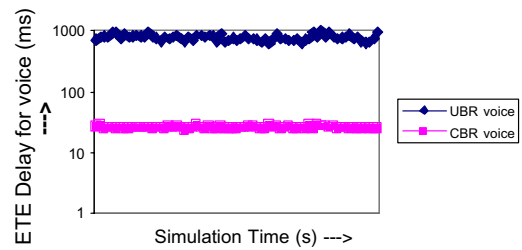


Fig. 7. ETE delays for UBR and CBR voice.

4.2. VoIP model

4.2.1. Weighted fair queuing

We first simulate our VoIP model for the traditional best-effort service offered by IP. Most of the current DSL deployments adopt ATM over DSL. Hence, in our initial experiments, we use ATM as the Layer 2 protocol. In our experimental setup, we have two end-hosts in the customer premises participating in voice calls. The G.729 encoder is used, with 1 frame per voice packet. The bandwidth requirement of a voice call is 42.4 kbps, including protocol headers. Data traffic is generated by an end-host uploading a 2 MB file via FTP. In the traditional best-effort IP model, there is no service differentiation between voice and data packets, both of which share the same queue. The FTP application generates a large amount of data traffic that severely degrades the performance of

Table 1
Comparison of VoATM and VoIP

Scenario	# Voice calls generated	# Voice frames per packet	Bandwidth per call including header (Kbps)	Upstream b/w reserved for voice (Kbps)	Up-stream interface MTU (bytes)	# Voice calls admitted	Mean ETE delay for voice (s)	ETE delay 95% confidence interval (s)
VoATM UBR voice	4	2	21.2	–	9180	4	0.663	(0.661, 0.665)
VoATM CBR voice	4	2	21.2	64	9180	3	0.0256	(0.0255, 0.0257)
VoIP-ATM@L2 No QoS	2	1	42.4	–	9180	2	6.55	(6.47, 6.63)
VoIP-ATM@L2 WFQ	2	1	42.4	89.6	9180	2	0.0777	(0.0770, 0.0783)
VoIP-ATM@L2 WFQ + Frag	2	1	42.4	89.6	328	2	0.0239	(0.0238, 0.0240)
VoIP-SLIP@L2 WFQ + Frag	3	2	19.2	64	328	3	0.0146	(0.0145, 0.0147)

voice traffic. Voice packets experience long queuing delays as they are trapped behind several large data packets in the queue. The mean ETE delays of the voice packets are unacceptably high for real-time applications.

Next, we incorporate the TOS-based WFQ mechanism at the CPE and repeat the above experiment. Voice and data packets are identified by different TOS values and placed in different queues. (*In this experiment, we have just one category for voice packets. Later, when we describe our experiments with ACIS, we have premium voice and regular voice.*) Since we have two voice calls, each requiring 42.4 kbps, we must reserve at least $2 \times 42.4 = 84.8$ kbps upstream bandwidth for voice traffic. In our experiment, we assign weights to the queues such that 70% of the upstream bandwidth (70% of 128 kbps = 89.6 kbps) is reserved for voice traffic. The ETE delays for the voice packets drop considerably, because the voice packets now have a separate queue and are served in a timely fashion. The results are summarized in Table 1.

4.2.2. Fragmentation

Although WFQ succeeded in reducing the ETE delay of voice packets, WFQ alone is not good enough. Note that the ETE delay obtained in the WFQ experiment is about 78 ms, but typical access network delays for voice calls ought to be less than about 30 ms. Furthermore, the delay variation, though small, is still higher than what we observed in the VoATM with CBR experiment. The reason is that even though voice packets have their share of bandwidth, they cannot preempt a data packet that is already being transmitted when they arrive. The end-hosts on the customer premises run on Ethernet, which supports packet sizes of up to 1500 bytes. A 1500 byte packet takes roughly 94 ms to be transmitted over a 128 kbps link. Thus, an arriving voice packet may have to wait anywhere from 0 to 94 ms in the queue, depending on whether a data packet is already in transmission and how much of that transmission remains. It is easy to see that in this scenario, very often the waiting time could be unacceptably high for voice packets. Fragmentation of large data packets is expected to yield some improvement in performance. As discussed in Section 2, the use of PPP-

based fragmentation (rather than IP-based fragmentation) is beneficial for the fragmented packets. From the point of view of voice traffic, both IP-based and PPP-based fragmentation produce similar results, since voice packets never get fragmented.

We illustrate by simulating IP fragmentation. To investigate the performance improvement obtained by fragmentation of large data packets, we reduce the MTU of the upstream interface to a value below 1500 bytes. We choose a value such that the data packets are not too severely fragmented. At the same time, we wish to limit the queuing delay of arriving voice packets to acceptable levels. Caputo [7] suggests a figure of 20 ms as the upper limit for transmission time of packets on a low-speed link. A 128 kbps link can transmit $(128\,000 \times 20) / (8 \times 1000) = 320$ bytes of data in 20 ms. Hence, we select a value close to 320 bytes as the MTU of the upstream interface. Since the Layer 2 protocol is ATM, we choose the MTU such that the packet will fit exactly into an integer number of ATM cells. We thus eliminate any need for padding. In our experiment, we chose an MTU of 328 bytes. After adding an AAL5 trailer of 8 bytes, we get a total of 336 bytes, which then fits exactly into 7 ATM cells. Along with their headers, these 7 ATM cells make up $7 \times 53 = 371$ bytes of data. Thus, the maximum transmission time for a packet is $(371 \times 8) / 12\,800 = 23$ ms. We repeat the experiment for VoIP with WFQ after setting the MTU of the CPE's upstream interface to 328 bytes. The results (tabulated in Table 1) show that the fragmentation of large data packets into smaller packets results in lower delay and lower variations in delay experienced by voice traffic. The values for ETE delay obtained in this experiment are comparable to the values in the VoATM with CBR experiment. Fig. 8 illustrates the improvement in VoIP performance achieved through QoS mechanisms.

4.2.3. Comparison of layer 2 protocols

We now turn our attention to the bandwidth requirements for voice calls. Recall that in our VoIP experiments, each voice call used the G.729 encoder and required a bandwidth of 42.4 kbps. The G.729 encoder produces a raw voice bit

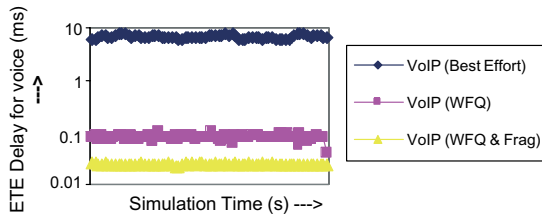


Fig. 8. Improvement in VoIP performance.

stream at the rate of 8 Kbps. The remaining bandwidth ($42.4 - 8 = 34.4$ kbps) is for the protocol headers. Voice over IP over ATM is extravagant in terms of bandwidth utilization. The problem with the IP over ATM protocol stack for voice is that voice packets should be small in order to minimize the packetization delay and the transmission delay (especially on low speed links). In the IP over ATM (AAL5) model, AAL5 adds an 8-byte trailer to the voice payload and the ATM layer adds a 5-byte header for every 48 bytes of payload. Since the cell sizes are fixed, the ATM layer may also add as much as 47 bytes of padding. All this is in addition to the UDP and IP headers. (Since the version of OPNET ModelerTM we use does not support RTP, our model does not include RTP. RTP of course would impose additional overhead.) As discussed in Section 1, techniques such as header compression and IP trunking alleviate the overhead problem, but an investigation of these techniques is beyond the scope of our research. Instead, we look at SLIP as an alternative to ATM in Layer 2. SLIP is used for exchange of data over point-to-point links (such as our ADSL link). SLIP has a very small header and imposes low protocol overhead. Moreover, the payload size of a SLIP frame is not fixed and hence there is no need for padding. Fig. 9 compares the bandwidth requirement of VoIP over ATM with that of VoIP over SLIP, when G.729 is used as the encoder.

The bandwidth requirement for VoIP over SLIP is consistently lower than that of VoIP over ATM. We also note that increasing the number of voice frames per packet (therefore increasing packetization delay) does not always translate into lower bandwidth requirements in the case of ATM. This is due to ATM's use of fixed-size cells, and consequent need for padding.

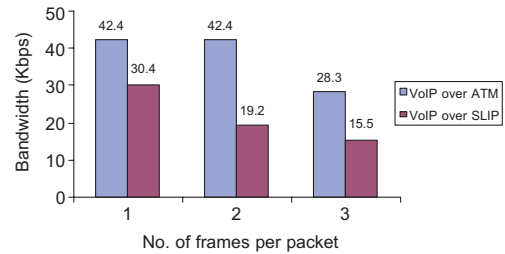


Fig. 9. Bandwidth requirement—VoIP over ATM, SLIP.

For the rest of the experiments described in this paper, we replace ATM with SLIP in Layer 2 because of its superior bandwidth efficiency. Currently, SLIP is being replaced by PPP, a more sophisticated protocol that accomplishes fundamentally the same tasks. We used SLIP in our experiments since OPNET ModelerTM 7.0 does not provide support for PPP. However, like SLIP, PPP also imposes a low protocol overhead. Hence, the bandwidth requirements for a voice call using VoIP with PPP at Layer 2 protocol will be similar to that of SLIP. In addition, PPP also allows us to take advantage of the fragmentation by Multi-link PPP, as described in Section 2.

We now describe an experiment that uses VoIP over SLIP. Three end-hosts in the customer premises are participating in voice calls and one end-host is uploading a 2 MB file using FTP. The end-hosts use the G.729 encoder and pack two frames in each voice packet. The bit rate of a voice call is 19.2 kbps. Hence, we need to reserve at least $19.2 \times 3 = 57.6$ kbps upstream bandwidth for the voice traffic. Note that when compared to the 'VoIP over ATM' experiment, we need to reserve less bandwidth in this case, even though we support an additional voice call. For this experiment, we reserve 50% of the upstream bandwidth (50% of 128 kbps = 64 kbps) for voice packets. The MTU of the upstream interface is set to 328 bytes. The results are shown in Table 1. Interestingly, the ETE delays of the voice packets for 'VoIP over SLIP' are the lowest that we have observed in our experiments. Recall that in VoATM and in 'VoIP over ATM,' the ETE delays were approximately 25 ms. We attribute this difference to the relative complexity of the ATM protocol stack when compared to SLIP. The packet processing functions at the

various layers of the ATM protocol stack cause the additional delay in the VoATM and ‘VoIP over ATM’ experiments. To verify this, we collected results for the ETE delay of the ATM cells and also the ETE delay of the AAL PDUs. We found that the ETE delays of the AAL PDUs were much higher than those of the ATM cells. This points to delay imposed by the ATM stack. Thus, the simplicity of SLIP not only offers greater bandwidth efficiency, but also has the potential to lower ETE delay.

4.3. ACIS

In all our VoIP experiments so far, we have computed the total bandwidth requirement for all the voice calls and reserved sufficient upstream bandwidth at the CPE. Of course the number of simultaneous voice calls may not always be known in advance. Without any form of admission control, it is possible for offered voice traffic load to exceed the bandwidth reserved for it, degrading the performance of all voice calls. This is illustrated in our next experiment.

Four end-hosts in the customer premises are engaged in voice calls. They use the G.729 encoder with 2 voice frames per packet. The protocol stack used is VoIP over SLIP. The resulting bit rate of a voice call is 19.2 kbps. Another end-host uploads a 2 MB file via FTP. The CPE implements both WFQ and fragmentation of large packets. Assume that the number of calls is not known in advance and that 35% of the upstream bandwidth (35% of 128 kbps = 44.8 kbps) is reserved for voice traffic. The bandwidth

requirement of the four voice calls is much higher $19.2 \times 4 = 76.8$ kbps. The results (tabulated in Table 2) show that the mean ETE delays of voice packets are unacceptably high in this scenario, since the FTP application generates a heavy load of competing data traffic and the bandwidth reserved for voice is insufficient.

To prevent such a situation from occurring, we implement the ACIS mechanism at the CPE. ACIS classifies voice calls into two categories—premium and regular. Both have a certain amount of upstream bandwidth reserved for them. However, the difference is that ACIS does not admit any voice calls as premium calls after all the premium bandwidth has been reserved by previous calls. We repeat the above experiment with 35% of the upstream bandwidth (35% of 128 kbps = 44.8 kbps) reserved for premium voice traffic and 15% of the upstream bandwidth (15% of 128 kbps = 19.2 kbps) reserved for regular voice traffic. Since only 44.8 kbps is reserved for premium voice, the CPE can admit only two voice calls, which will reserve $19.2 \times 2 = 38.4$ kbps. The remaining 6.4 kbps are not sufficient to accommodate a third call. Hence, the remaining two calls are admitted as regular calls. The results (refer Table 2) show that premium calls are ensured low ETE delays, while regular calls experience high ETE delays in the presence of competing FTP traffic. Fig. 10 shows the effectiveness of the ACIS mechanism is protecting premium voice calls.

We performed additional experiments with different traffic loads and different types of competing traffic, such as email and web browsing. The results obtained were fundamentally the same. For

Table 2
Performance of the admission control algorithm

Scenario	# Voice calls generated	Upstream b/w reservation for premium voice (Kbps)	# Voice calls admitted as premium	Mean ETE delay for regular voice (s)	ETE delay 95% confidence interval regular voice (s)	Mean ETE delay for premium voice (s)	ETE delay 95% confidence interval premium voice (s)
VoIP-SLIP@L2 No admission ctrl	4	–	–	0.0509	(0.0483, 0.0535)	–	–
VoIP-SLIP@L2 With admission ctrl	4	35	2	0.0658	(0.0600, 0.0717)	0.01355	(0.0135, 0.0136)

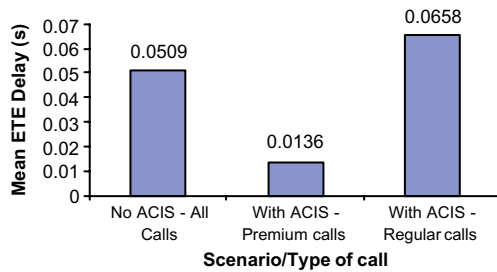


Fig. 10. Performance of ACIS.

details on these results, the reader is referred to [24].

5. Conclusions

The objective of our research was to evaluate the feasibility of using VoIP instead of VoATM in future DSL deployments. We have proposed and evaluated a VoIP architecture suitable for deployment in the DSL access network. Our results show that the performance of our architecture is comparable to that of the existing VoATM architecture. In addition, it also offers the benefit of voice packet compatibility with the core network, which is already employing VoIP to carry voice inexpensively over data networks. This is a positive step towards widespread deployment of end-to-end IP telephony.

As part of our architecture, we proposed a new admission control mechanism called Admission Control by Implicit Signaling (ACIS). We advocate the use of ACIS as it eliminates the need for an explicit signaling protocol for admission control at the CPE. Instead, it takes advantage of existing application layer signaling protocols by mapping them to the network layer. Our results show that ACIS is capable of providing excellent QoS to premium calls at all loads and acceptable QoS to regular calls at light or medium loads. Since the typical ADSL user runs applications with low loads in the upstream direction, there is a fair chance that the regular calls receive an acceptable performance.

We also illustrated how using ATM at Layer 2 imposes an extremely high protocol overhead on

voice packets. Moreover, VoIP does not need to rely on the use of ATM QoS. Thus we propose that future deployments of VoIP eliminate ATM altogether in favor of a lightweight protocol like PPP. The use of PPP at Layer 2 would also enable us to take advantage of fragmentation by Multi-link PPP.

IP QoS mechanisms are now reaching maturity. In this paper, we describe how to enable such mechanisms without the need for an additional signaling protocol in a DSL environment. This in turn yields acceptable quality in the delivery of voice over the access network, often the major bottleneck.

Future research in this area would include a more detailed assessment of the performance of data traffic in our VoIP architecture. This would involve a study of TCP's ability to efficiently utilize the residual bandwidth (bandwidth unused by voice traffic) in the link. The coexistence of video, voice and data traffic over DSL can also be considered.

Currently, ACIS is designed for the specific task of admission control of voice calls over the upstream ADSL link. ACIS can be extended to work in conjunction with MBAC or a bandwidth broker. This would enable the CPE to vary the bandwidth reserved for premium calls dynamically. This could be done based on measurements of the traffic load or by consulting a bandwidth broker.

Acknowledgements

A condensed preliminary version of this article appears in the proceedings of IEEE GLOBECOM 2003.

References

- [1] G. Armitage, *Quality of Service in IP Networks*, MacMillan, New York, 2000, pp. 169–183.
- [2] G. Bianchi, A. Capone, C. Petrioli, Throughput analysis of end-to-end measurement-based admission control in IP, in: *Proceedings INFOCOM 2000*, vol. 3, 2000, pp. 1461–1470.
- [3] G. Bianchi, N. Blefari-Melazzi, M. Femminella, F. Pugini, Performance evaluation of a measurement-based algorithm

- for distributed admission control in a DiffServ Framework, in: Proceedings GLOBECOM 2001, vol. 3, 2001, pp. 1886–1891.
- [4] F. Borgonovo, A. Capone, L. Fratta, M. Marchese, C. Petrioli, PCP: a bandwidth guaranteed transport service for IP networks, in: Proceedings ICC 1999, vol. 1, 1999, pp. 671–675.
- [5] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification, IETF RFC 2205, September 1997.
- [6] L. Breslau, E.W. Knightly, S. Shenker, I. Stoica, H. Zhang, Endpoint admission control: architectural issues and performance, in: Proceedings ACM SIGCOMM 2000, 2000, pp. 57–69.
- [7] R. Caputo, Cisco Packetized Voice and Data Integration, McGraw-Hill, New York, 1999, pp. 64–109.
- [8] S. Casner, V. Jacobson, Compressing IP/UDP/RTP headers for low-speed serial links, IETF Internet Draft, 1998.
- [9] CopperCom Technology White Paper, Mastering Voice Over DSL: Network Architecture.
- [10] CopperCom Technology White Paper, Seven Essentials of a Local Exchange Softswitch System, Part 4: Packet Voice over Broadband.
- [11] Copper Mountain Networks White Paper, Voice Over DSL.
- [12] A. Demers, S. Keshav, S. Shenker, Analysis and simulation of a fair queuing algorithm, in: Proceedings ACM SIGCOMM 1989.
- [13] A. Detti, M. Listanti, S. Salsano, L. Veltri, Supporting RSVP in a differentiated service domain: an architectural framework and a scalability analysis, in: Proceedings ICC 1999, vol. 1, 1999, pp. 204–210.
- [14] V. Elek, G. Karlsson, R. Ronngren, Admission control based on end-to-end measurements, in: Proceedings INFOCOM 2000, vol. 2, 2000, pp. 623–630.
- [15] K. El-Khatib, G. Luo, G. Bochmann, Multiplexing scheme for RTP flows between access routers, IETF Internet Draft, 1999.
- [16] Y. Hou, Z. Duan, Z. Zhang, T. Chujo, Providing scalable support for multiple QoS guarantees: architecture and mechanisms, in: Proceedings ICC 2001, vol. 7, 2001, pp. 2115–2122.
- [17] V. Jacobson, Compressing TCP/IP headers for low speed serial links, IETF RFC 1144, 1990.
- [18] Y. Jia, M. Chen, A novel architecture of providing guaranteed services for differentiated services network, in: Proceedings EUROCON 2001, vol. 2, 2001, pp. 492–495.
- [19] R. Mamelì, S. Salsano, Use of COPS for Intserv operations over DiffServ: architectural issues, protocol design and Test-bed implementation, in: Proceedings ICC 2001, 2001, pp. 3265–3270.
- [20] K. Mase, Y. Toyama, A.A. Bilhaj, Y. Suda, QoS management for VoIP networks with edge-to-edge admission control, in: Proceedings GLOBECOM 2001, vol. 4, 2001, pp. 2556–2560.
- [21] J. Mogul, S. Deering, Path MTU discovery, IETF RFC 1191, 1990.
- [22] R. Mortier, I. Pratt, C. Clark, S. Crosby, Implicit admission control, IEEE Journal on Selected Areas in Communications. 18 (2) (2000) 2629–2639.
- [23] W. Qian, W. Jing, C. Shiduan, M. Jian, A PHB reservation mechanism in Diffserv network, in: Proceedings International Conference on Communications Technology (ICCT 2000), vol. 1, 2000, pp. 424–427.
- [24] A. Ram, Assessment of voice over IP as a solution for voice over ADSL, M.S. Thesis, Virginia Polytechnic Institute and State University, May 2002.
- [25] W. Rhee, H. Kim, K. Park, K. Kim, S. Kim, Two phase edge-to-edge distributed measurement based admission control mechanism in large IP networks, in: Proceedings GLOBECOM 2001, vol. 4, 2001, pp. 2571–2575.
- [26] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, SIP: session initiation protocol, IETF RFC 2543, 1999.
- [27] W. Simpson, The point-to-point protocol (PPP), STD 51 IETF RFC 1661, 1994.
- [28] K. Sklower, B. Lloyd, G. McGregor, D. Carr, T. Coradetti, The PPP multilink protocol (MP), IETF RFC 1990, 1996.
- [29] I. Stoica, H. Zhang, Providing guaranteed services without per flow management, in: Proceedings ACM SIGCOMM 1999, 1999, pp. 81–94.
- [30] K. Tajima, M. Hashmani, M. Yoshida, A resource management architecture over differentiated services domains for guarantee of bandwidth, delay and jitter, in: Proceedings EUROCOM 2000, 2000, pp. 242–249.
- [31] TollBridge Technologies White Paper, TollVoice Architecture and Local Exchange Gateways: Enabling Technologies for Voice over Broadband.
- [32] F. Wang, P. Mohapatra, An efficient bandwidth management scheme for real-time Internet applications, in: Proceedings International Symposium on Intelligent Multimedia, Video and Speech Processing, 2001, pp. 469–472.
- [33] G. Zhang, H.T. Mouftah, End-to-end QoS guarantees over Diffserv, in: Proceedings 6th IEEE Symposium on Networks, Computers and Communications, 2001, pp. 302–309.



Abhishek Ram hails from Chennai, India. He earned his Bachelor's degree in Computer Engineering at Karnataka Regional Engineering College, Surathkal, India and his Master's degree in Computer Science at Virginia Tech, Blacksburg, Virginia. He currently works as a Software Design Engineer in the Windows Device Experience Group at Microsoft Corporation, Redmond, Washington.



Luiz A. DaSilva joined Virginia Tech as an Assistant Professor at the Bradley Department of Electrical and Computer Engineering in 1998, after receiving his Ph.D. in Electrical Engineering at the University of Kansas. He has previously worked for IBM for six years. His research interests focus on performance and resource management in wireless mobile networks and Quality of Service (QoS) issues. Current and recent research sponsors include NSF, the Office for Naval Research, the US Customs Services, Intel, and Microsoft Research, among others. He is a member of the Center for Wireless Telecommunications (CWT), associated faculty at the Mobile and Personal Radio Research Group (MPRG), and a member of the Governing Board of the NSF-funded Integrated Research and Education in Advanced

Networking (IREAN) program at Virginia Tech. He is a senior member of IEEE and a member of ASEE.



Srinidhi Varadarajan is an assistant professor of computer science and the director of the Terascale Computing Facility (TCF) at Virginia Tech. His research is focused on scalable network emulation, compiler-directed frameworks for flexible data sharing and application composition, and routing algorithms for backbone IP networks. He is the recipient of a 2002 US National Science Foundation CAREER award, the 2003 New Century Technology Council Innovation award, and was named a 2003 faculty fellow by the College of Engineering, Virginia Tech. He holds a Ph.D. in Computer Science from the State University of New York, Stony Brook.